Intelligent Monitor for an Anesthesia Breathing Circuit

Scott P. Narus, M.S., Kai Kück, Dwayne R. Westenskow, Ph.D.,
Anesthesiology Bioengineering Laboratory, University of Utah,
Salt Lake City, Utah

A competent breathing circuit is mandatory to the safe and effective delivery of oxygen and anesthetic gases to the patient. Studies have shown that failures in the circuit are the most likely causes of anesthetic mishaps. Unfortunately, the complexity of the system renders traditional monitoring methods ineffective. We have developed a hierarchical artificial neural network monitor that is capable of examining ventilator signals. It was trained to identify 23 faults in the breathing circuit during ventilator controlled breathing and 21 faults during spontaneous breathing. The networks correctly identified a fault condition in 92% and 83% of cases for ventilator and spontaneous data, respectively. The correct fault type was found in 76% and 68% of cases for ventilator and spontaneous data, respectively. Results show that the network met our criteria for a holistic, specific, and vigilant monitoring system.

INTRODUCTION

During surgical procedures, a patient who receives general anesthesia will require breathing assistance from a mechanical ventilator in order to accommodate for the artificially suppressed pulmonary drive. The ventilator regulates respiratory rate, tidal volume, and gas mixture in order to provide oxygen to the patient and to remove CO_2. Additionally, if inhaled anesthetic gases are used, the ventilator provides these agents in the proper concentrations.

The most common breathing circuit used for anesthesia in the United States is the circle breathing circuit, a semi-closed system that chemically removes CO_2 from exhaled gases, scavenges excess anesthetic gases, and adds fresh air to replace any lost volume. Although simple in design, the breathing circuit is mechanically complex because of the many hoses, fittings, valves and drive mechanisms required. This complexity can lead to failures in the breathing circuit due to disconnections, obstructions, leaks and incompetent valves. All of these failures can have serious consequences for the patient. As several studies of anesthetic mishaps have shown, most respiratory mishaps are due to failures in the ventilator circuit[1-3] . It is believed that most of these could be avoided through better monitoring.

Traditionally, threshold alarms have been used on breathing circuits to monitor for out of range CO_2, pressure, and/or flow. Threshold alarms have limited capabilities, though, because they typically do not automatically adjust for dynamic changes in either the patient or the breathing circuit. They also are extremely limited in the amount of information that they can convey to the anesthetist because any single threshold alarm does not distinguish multiple etiologies of malfunction (e.g. a "high pressure" alarm could indicate a hose obstruction, a stuck valve or a change in patient physiology). A specific malfunction is not identified; the clinician must seek the cause of the alarm, often unsystematically and unsuccessfully. Threshold alarms are also notorious for causing numerous "false" alarms, unfortunately forcing many users to simply disable them.

A better breathing circuit monitor will be holistic, specific, and vigilant. By holistic we mean that the monitor will simultaneously examine many aspects of the system and use several pieces of information in order to determine the state of the system. By specific we mean that the monitor will provide a distinct message for each breathing circuit failure it is capable of detecting. Finally, by vigilant we mean that the monitor will constantly scrutinize the system throughout the entire surgical procedure, adjusting to changes in the operating environment that may affect the monitor's performance.

To implement a monitor with these capabilities, the detection system must be able to quickly process diverse signals from a wide range of operating environments, and then resolve circuit failures that may have only subtle distinctions. Because anesthesiologists often do not recognize abnormalities in available waveform data that may indicate many of the important circuit faults, the monitoring system must be able to independently discover underlying relationships between input signals and circuit status. Artificial neural networks (ANNs) can provide the capabilities described in order to implement an intelligent monitoring system.

METHODS

Background

The Anesthesiology Bioengineering Laboratory, in conjunction with Ohmeda, has been developing an ANN-based alarm system for the Ohmeda Modulus CD anesthesia machine. The alarm system uses CO_2, pressure and flow waveform signals provided by the machine in order to detect 23 faults in the breathing circuit (see Table 1).

Orr[4] used an ANN to process signals from airway CO2, pressure, and flow sensors in order to detect 14 faults in the breathing circuit. The input signals were collected using a Michigan Instruments lung simulator with the ventilator in drive mode. Results of an ANN for detecting 19 faults using the Modulus CD's commercially equipped CO_2, pressure and flow sensors were reported by Farrell et. al[5]. They collected data from seven dogs anesthetized with halothane and mechanically ventilated. Several different flow rate, tidal volume, and inspiratory: expiratory ratio settings were used, along with different combinations of lung compliance and airway resistance. Differential features were used in order to simplify detection and differentiate fault conditions from system recovery. The research presented here extends this system to 23 faults, and adds the capability to detect faults during simulated spontaneous breathing. Absolute features were used in this research so that the state of the breathing circuit could be determined immediately. In addition, the use of hierarchical network structures were studied in order to optimize the monitoring system.

Data Collection

Two sets of data were collected for this research. For the first data set, signals were generated during ventilator controlled breathing. The Ohmeda anesthesia machine was attached to the Michigan Instruments Ventaid test lung. Tidal volume, respiratory rate, inspiratory to expiratory ratio (I:E ratio) and fresh gas flow were varied to provide a wide range of operating conditions. Lung compliance and airway resistance were also varied. The combinations of all the ventilator settings, lung compliances and airway resistances produced 48 different operating conditions for each of the 23 faults shown in Table 1.

Faults in the breathing circuit were manually created by disconnecting hoses, pinching hoses, opening pre-made "leaks", and clamping one-way valves. Each fault was maintained for at least five breaths. The fault was then removed and the system allowed to return to its normal operating level before the next fault was created. The breaths generated during recovery from a fault, until the time when the system returned to a normal level, were collected but not used for ANN training. Each fault constituted a numbered "event", regardless of how many breaths it lasted. For the 23 faults and 48 operating conditions, this came to 1104 total fault events, plus 1105 non-fault (normal) events.

CO_2, flow and pressure signals from transducers available on the Modulus CD were sampled at 30 Hz by

Table 1. Breathing Circuit Faults

1. Leak in endotracheal tube
2. Leak in inspiratory hose
3. Leak in CO_2 canister
4. Gas sampling port disconnection
5. Fresh gas hose disconnection
6. Endotracheal tube disconnection
7. Disconnect at Y-piece, distal to sampling port
8. Inspiratory hose disconnection
9. Expiratory hose disconnection, proximal
10. Expiratory hose disconnection, distal
11. Ventilator hose disconnection
12. Expiratory hose obstruction
13. Leak in expiratory hose
14. Inspiratory hose obstruction
15. Endotracheal tube obstruction
16. Inspiratory valve stuck open
17. Inspiratory valve stuck closed
18. Expiratory valve stuck open
19. Expiratory valve stuck closed
20. O_2 sensor disconnect
21. TVX clip off
22. Pressure sensor line occluded
23. Leak in bellows

a personal computer attached to the anesthesia machine. A breath detection algorithm was used on each of the three signals: when a full breath was detected for a particular signal, a set of features was derived from the signal and saved to computer disk, along with header information to identify the ventilator state, fault condition, event number, breath number, lung compliance and airway resistance. The occurrence of a breath detection for any one of the three signals constituted a numbered "block". Sixteen CO_2 features, 14 flow features, and 14 pressure features were calculated, respectively.

A second set of "spontaneous" respiratory data was collected from the Ohmeda machine. Spontaneous breathing was simulated by driving one side of the lung simulator with a second ventilator while the other side provided signals to the anesthesia machine. The second ventilator was controlled by a computer program that randomly varied inspiratory pause, tidal volume, respiratory rate and inspiratory flow around a set baseline. A novel automatic fault creator (developed by the Anesthesia Bioengineering Laboratory) was used to generate the circuit defects, eliminating any human inconsistency. Because this second data set simulated spontaneous breathing, "ventilator hose disconnection" (fault 11) and "leak in bellows" (fault 23) were eliminated from the data set. "TVX clip off" (fault 21) was also eliminated because it could not be automatically created. This left 20 faults against which to train the spontaneous ANN.

ANN Modularization

The ANN hierarchy consisted of three stages: Stage 1 was a single network that classified pulmonary compliance and resistance (four possible values: "high-high", "high-low", "low-high", "low-low"); Stage 2 had four networks (one for each compliance-resistance type) that determined if a fault was present; Stage 3 had four networks (one for each compliance-resistance type) that determined the fault type (23 possibilities for ventilator controlled breathing or 20 for spontaneous breathing). This hierarchical structure is shown in Figure 1.

It was hypothesized early in ANN development that separating data sets based on lung compliance and airway resistance would provide both better fault resolution and an added piece of physiological data for the clinician. Therefore, an initial stage was added to the ANN hierarchy that determined lung compliance ("high" or "low") and airway resistance ("high" or "low"). In our test configuration, high compliance and low resistance were considered "normal" conditions.

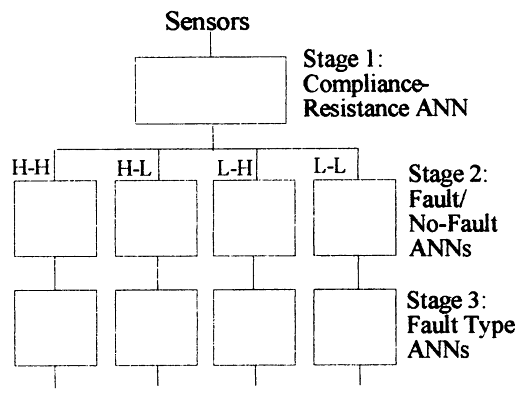
A separate fault/no fault ANN was trained for each of the four compliance-resistance types. The result of the compliance-resistance ANN was used to select the proper ANN. This hierarchy more than quadrupled the number of ANNs, but each ANN could be trained with one-fourth the data of a monolithic ANN, and the network complexity (i.e. number of middle layers and nodes per middle layer) could be reduced.

During data collection, many more normal condition blocks were generated than fault condition blocks. ANNs are susceptible to slow training and biased outcomes if the distribution of target classes is not uniform.[6] In order to use as many normal blocks as possible, with the hope of lowering false positive rates, the system was divided into a preliminary stage that determined the fault/no fault condition, and a secondary stage that discriminated the fault type if the preliminary stage indicated a breathing circuit fault. Thus, the preliminary stage ANN was able to use as many normal blocks as there were total fault blocks in a training set. The secondary stage ANN was able to use an equivalent number of blocks for each fault type in its training set. This eliminated any bias in either stage and provided the maximum number of training patterns.

Data Preprocessing

Because the ANN processing was divided into three stages, three different training data sets were needed.

Figure 1. ANN Architecture



First, the number of blocks for each fault was found. Then, the fault with the least number of blocks, N_{min}, was identified. An 80/20 training pattern to testing pattern ratio was used. Therefore, the number of training blocks for any fault was $0.8 \bullet N_{min}$. A master training data file was generated by randomly picking $0.8 \bullet N_{min}$ patterns of each fault type, along with $23 \bullet 0.8 \bullet N_{min}$ normal blocks or $20 \bullet 0.8 \bullet N_{min}$ normal blocks (randomly chosen) for the ventilator controlled breathing set and the spontaneous breathing set, respectively. The remaining patterns were held out to construct a master test data file.

All the patterns from the master training file were used to train the Stage 1 network. The patterns were normalized by first finding the median and median deviation (absolute deviation from the median) of each feature. Median and median deviation were used because feature distributions were rarely Gaussian and often contained significant outliers. Thus, the median and median deviation provided a more robust feature set than that given by mean/standard deviation or maximum/minimum normalization techniques. To normalize each feature of each pattern, the feature's median was subtracted from the feature and the difference was then divided by the median deviation. The master test file was normalized using these same values.

For Stage 2, four training and testing data files were constructed, one for each of the four networks. Each file contained only those patterns that corresponded with the compliance/resistance type. The median and median deviation of each training file were used to normalize the data.

The files from Stage 2 were used to form the Stage 3 files by simply deleting the normal patterns. Once again, the median and mean deviation of each training file were used to normalize the data.

We used standard "one output node per class" coding to generate the target vectors. The node corresponding to the target class was set to 0.9 while the other nodes were set to 0.1. For the Stage 1 ANN, this produced a vector of size 4. For the Stage 2 ANNs, we were able to use a single output node since coding with two nodes is the mathematical equivalent. For the Stage 3 ANNs, we had vectors of size 23 and of size 20 for ventilator controlled breathing and spontaneous breathing, respectively.

ANN Training

Multilayer, feedforward perceptron networks (MLPs) were used exclusively in this research. All nodes in active layers used logistic activation functions with bias inputs. Each layer in a network was fully connected to its adjacent layers. We used three layer networks in all stages. The first layer was a pass-through layer that simply directed all features in unmodified form to all the nodes in the second layer. Various second layer sizes were tried before settling on an optimum configuration. For Stage 1, this turned out to be 20 nodes; For Stage 2, 15 nodes; For Stage 3, 30 nodes.

Each ANN in the hierarchy was trained using backpropagation with momentum. Weights were updated after each pattern presentation. An internal test set, consisting of 10% of the patterns in a training set, was held out and used to judge when to stop training: training was stopped when mean squared error on the internal test set began to consistently diverge from that of the remaining training set. The network was then evaluated against the corresponding test data set.

RESULTS

Table 2 shows the percent of correctly identified blocks for the Stage 1 compliance-resistance networks for the ventilator controlled and spontaneous breathing data sets. Uncharacteristically, results for the test sets are slightly better than those for the training sets. The test sets contained a much higher percentage of normal blocks, which likely biased the results. Surprisingly, we found that the network trained with the spontaneous breathing data performed as well as the network trained with the ventilator controlled breathing data.

Table 2. Results for Stage 1 Networks

	Ventilator		Spontaneous	
	Train	Test	Train	Test
Correct (%) (N)	92 (25446)	97 (21301)	92 (31550)	97 (25058)

The results for the Stage 2 fault/no-fault networks are shown in Table 3. The percent of True Positive (TP) and False Positive (FP) blocks are given for the networks trained to recognize high compliance-low resistance blocks. Results for the other Stage 2 networks were nearly identical. As expected, fault detection (TP) during ventilator controlled breathing was superior to that of spontaneous breathing.

Table 3. Results for Stage 2 H-L Networks

	Ventilator		Spontaneous	
	Train	Test	Train	Test
TP (%) (N)	93 (3201)	92 (809)	86 (3935)	83 (992)
FP (%) (N)	1 (3201)	3 (4982)	0 (3935)	4 (5093)

We also examined how long it took each network to correctly identify a fault condition. Table 4 shows the percent of correctly identified fault blocks at each breath after the fault was created (results shown are for the high compliance/low resistance network). Note that by the third breath virtually all the blocks are correctly identified by the ventilator network. The spontaneous network appears to need at least five breaths before it operates at a nearly equal level.

Table 4. Percent of Correctly Identified Fault Blocks at the Corresponding Breath After Fault Creation

		Breath Number				
		1	2	3	4	>4
Ventilator						
Train	% (N)	56 (391)	93 (590)	100 (563)	99 (571)	100 (1086)
Test	% (N)	54 (98)	93 (130)	99 (152)	98 (139)	99 (290)
Spontaneous						
Train	% (N)	20 (329)	69 (445)	87 (447)	90 (439)	97 (2275)
Test	% (N)	18 (84)	58 (103)	87 (109)	90 (107)	95 (589)

Although the Stage 2 networks missed 8% and 17% of the fault blocks, we were encouraged that the results of Table 4 might show that, given enough blocks, every fault could be detected. Therefore, we developed a criterion that stated that if 3 of the last 5 blocks indicated a fault, then a fault must exist. Using this criterion, the networks were able to find every fault event in the ventilator controlled breathing data set, while missing only two events in the spontaneous breathing data set. We were also able to use this criterion to lower false positive indications.

Table 5 shows the results of the Stage 3 fault discriminator networks, again for the high compliance-low resistance networks. This was the most difficult test for the ANNs. However, the ventilator network correctly identified 80% of the fault patterns, while the spontaneous data network correctly identified 75%. For the ventilator network, there were only 9 faults with blocks incorrectly classified more than 20% of the time, while 3 of the faults were incorrectly identified less than 10% of the time. For the spontaneous network, there were 9 faults with blocks incorrectly identified more than 25% of the time, while 4 of the faults were incorrectly identified less than 20% of the time.

Table 5. Results for the Stage 3 H-L Networks

	Ventilator		Spontaneous	
	Train	Test	Train	Test
Correct (%)	86	80	83	75
(N)	(3024)	(3986)	(2664)	(2263)

DISCUSSION

We have demonstrated that a hierarchical artificial neural network can correctly identify faults in an anesthesia breathing circuit in 83% or more of the cases. The network can distinguish the fault type in 75% or more of the cases. The network also provides valuable clinical data by correctly determining lung compliance and airway resistance at least 92% of the time. It is able to work in both ventilator controlled and spontaneous breathing modes. The network's abilities to simultaneously monitor several aspects of the breathing circuit, provide exact fault location messages, and diligently evaluate the circuit throughout an anesthetic case therefore meets our criteria for a holistic, specific, and vigilant monitoring system.

Results reported here are for *blocks* of data. Typically, a CO_2 block, a flow block, and a pressure block made up one breath. While one block of data may be corrupted or may not contain enough information alone to distinguish a fault condition, a full breath's worth of data is likely to contain a better picture of the true state of the system. Therefore, an output decision rule that uses several blocks of data should improve classification results. This hypothesis was partially confirmed by the results shown in Table 4 and by the results of the 3-of-5 criterion discussed earlier.

Because the outputs of our Stage 1 and Stage 2 networks had low dimensionality, the ANNs were fully capable of achieving low output errors. However, high output dimensionality, as in the Stage 3 networks, hinders network training and may prevent convergence to suitable network parameters. Thus, we may need to define a fourth stage in the hierarchy that divides faults into logical clusters (i.e. disconnects, leaks, occlusions). We may also be able to employ error-correcting output codes instead of our "one node per class" coding in order to improve our fault classification.[7]

Future research goals include expanding the ranges of ventilator settings, lung compliances and airway resistances. While the operational range over which we tested was quite wide, we cannot positively predict performance outside this range. We also plan to test the networks against patients undergoing anesthetic procedures. This should test the robustness of our networks and provide us with additional network training data.

References

1. Orkin FK. Anesthetic Systems. *In* RD Miller, Ed., "Anesthesia: 2nd Edition" Vol.1, pp. 117-160, New York: Churchill Livingston. 1986.
2. Whitcher C, *et. al.* Anesthetic Mishaps and the Cost of Monitoring: A Proposed Standard for Monitoring Equipment. *J Clin Mon.* 1988; 4(1):5-15.
3. Caplan RA, Posner KL, Ward RJ, Cheney FW. Adverse Respiratory Events in Anesthesia: A Closed Claims Analysis. *Anesthesiology.* 1990; 72: 828-833.
4. Orr JA. An Anesthesia Alarm System Based on Neural Networks. Ph.D. Thesis, University of Utah. 1991.
5. Farrell RM, Orr JA, Kuck K, Westenskow DR. Differential Features for a Neural Network Based Anesthesia Alarm System. *Biomed Sci Inst.* 1992; 28: 99-104.
6. Ohno-Machado L. Identification of Low Frequency Patterns in Backpropagation Neural Networks. *In* JG Ozbolt, Ed., "Proceedings of the 18th Annual Symposium on Computer Applications in Medical Care," pp. 853-859, Washington D.C., 1994.
7. Dietterich TG, Bakiri G. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *J Artificial Intelligence Research.* 1995;2:263-286.